Assignment-4.2

Name: Mohammed Irfan
Hall Ticker No: 2303A51997
Batch: 15

Task Description-1

• Zero-shot: Prompt AI with only the instruction. Write a Python function to determine whether a given number is prime

```python
#write a python code to check whether the given number is prime or not
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True

number = int(input("Enter a number: "))
if is_prime(number):
    print(f"{number} is a prime number.")
else:
    print(f"{number} is not a prime number.")
```

Output:

```
OneDrive/Desktop/ai assistance coding/assignment-4.2.py"
Enter a number: 21
21 is not a prime number.
```

Task Description-2

• One-shot: Provide one example: Input: [1, 2, 3, 4], Output: 10 to help AI generate a function that calculates the sum of elements in a list.

```python
#write a python function to calculate the sum of the element in the list ,use this example input:[1,2,3,4]
#  and output need to be Output:10
def sum_of_list(elements):
    return sum(elements)
example_list = [1, 2, 3, 4]
result = sum_of_list(example_list)
print(f"Output: {result}")
```

Output:

```
Output: 10
```

Task Description-3

• Few-shot: Give 2–3 examples to create a function that extracts digits from an alphanumeric string.

```python
#write the python function the extract digits from a alphanumeric strings with examples
# a)input:"a1b2c3d4" output:[1,2,3,4]
# b)input:"abc123xyz" output:[1,2,3]
# c)input:"no_digits_here" output:[]
def extract_digits(alphanumeric_string):
    digits = [int(char) for char in alphanumeric_string if char.isdigit()]
    return digits
# Example inputs and outputs
examples = [
    "a1b2c3d4",
    "abc123xyz",
    "no_digits_here"
]
for example in examples:
    print(f'Input: "{example}" Output: {extract_digits(example)}')
```

Output:

```
Input: "a1b2c3d4" Output: [1, 2, 3, 4]
Input: "abc123xyz" Output: [1, 2, 3]
Input: "no_digits_here" Output: []
```

Task Description-4

• Compare zero-shot vs few-shot prompting for generating a function that counts the number of vowels in a string.

```python
# write a python code to count vowels in a string
def count_vowels(input_string):
    vowels = "aeiouAEIOU"
    count = sum(1 for char in input_string if char in vowels)
    return count


input_string = input("Enter a string: ")
vowel_count = count_vowels(input_string)

print(f"Number of vowels in the string: {vowel_count}")

#write a python function that counts the number of vowels in a string with example
#  a)input:"Hello World" output:3
#  b)input:"Python Programming" output:4
#  c)input:"Data Science" output:5

def count_vowels(input_string):
    vowels = "aeiouAEIOU"
    count = sum(1 for char in input_string if char in vowels)
    return count
# Example inputs and outputs
examples = [
    "Hello World",
    "Python Programming",
    "Data Science"
]
```

Explanation:-

```
'''
In the first prompt, the model was only given instructions, so it had to infer the logic for identifying
vowels on its own. In the second prompt, sample inputs and outputs were provided, which helped the model
clearly understand the pattern and expected behavior. The examples guided the model toward a more
 structured and reliable response, showing how providing examples improves output consistency.
'''
```

Output:

```
Enter a string: aei21o
Number of vowels in the string: 4
```

Task Description-5

• Use few-shot prompting with 3 sample inputs to generate a function that determines the minimum of three numbers without using the built-in min() function

```python
#Write a function that finds the minimum of three numbers without using the built-in min() function.

#Input: a = 3, b = 7, c = 5
#Output: 3

#Input: a = 10, b = 2, c = 8
#Output: 2

#Input: a = 6, b = 9, c = 4
#Output: 4
def find_min_of_three(a, b, c):
    if a <= b and a <= c:
        return a
    elif b <= a and b <= c:
        return b
    else:
        return c
# Example inputs and outputs
examples = [
    (3, 7, 5),
    (10, 2, 8),
    (6, 9, 4)
]
for a, b, c in examples:
    print(f'Input: a = {a}, b = {b}, c = {c} Output: {find_min_of_three(a, b, c)}')
```

Output:

```
Input: a = 3, b = 7, c = 5 Output: 3
Input: a = 10, b = 2, c = 8 Output: 2
Input: a = 6, b = 9, c = 4 Output: 4
```